

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-50278

(P2000-50278A)

(43) 公開日 平成12年2月18日 (2000.2.18)

(51) Int.Cl.

識別記号

F I

テ-マ-ト* (参考)

H 0 4 N 7/32

H 0 4 N 7/137

Z 5 C 0 5 9

審査請求 未請求 請求項の数12 O L (全 19 頁)

(21) 出願番号 特願平10-212378

(22) 出願日 平成10年7月28日 (1998.7.28)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 安藤 裕司

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(74) 代理人 100082762

弁理士 杉浦 正知

Fターム(参考) 5C059 KK19 LB05 MA00 MA04 MA05

MA23 MC11 MC38 MD01 NN02

NN03 NN28 PP05 PP06 PP07

PP16 SS11 SS20 UA02 UA33

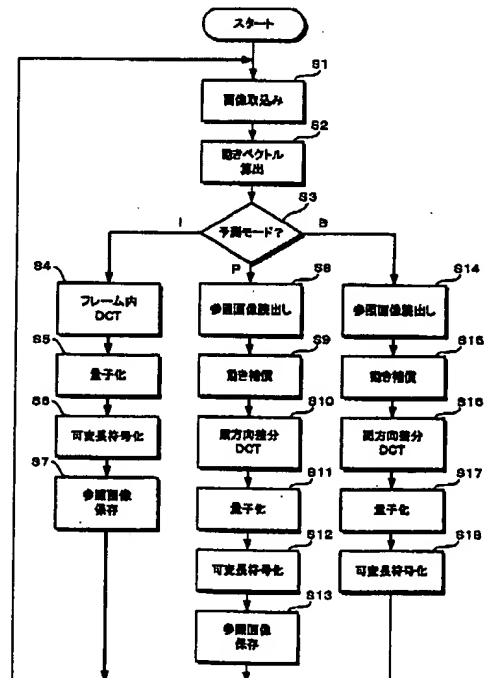
UA38

(54) 【発明の名称】 動きベクトル算出方法及び動きベクトル算出プログラムを記録した記録媒体

(57) 【要約】 (修正有)

【課題】 MPEG2のエンコードをソフトウェアで行う際に、探索領域内でのブロックマッチングの回数を減らして高速処理を図ると共に、MMX命令を有効に活用できるようにする。

【解決手段】 参照フレーム及び現フレームのブロック内の画素を市松模様状に間引いてブロックマッチングを行なう際に、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶する。これにより、ブロックマッチングの際にMMX命令を有効に利用することができ、高速化が図れる。また、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶すると、2画素ステップでサーチしていくことになるため、ログリズミックサーチが簡単に行なえる。



【特許請求の範囲】

【請求項 1】 処理の対象となる現画面において分割された現画面のブロックに対して、参照画面内における同位置を基点としてそのブロックと大きさの等しい参照画面のブロックを抽出し、

所定の探索範囲内で上記参照画面のブロックを動かしながら、上記現画面のブロックと参照画面のブロックとの間の残差を求め、

上記残差が最小となる参照画面のブロックを検出して動きベクトルを算出する動きベクトル算出方法において、上記現画面の画素及び上記参照画面の画素を N 画素 (N は整数) 毎に取り出し、

上記 N 画素毎に取り出された現画面の画素及び上記参照画面の画素を連続して並ぶデータとしてメモリに記憶させ、

上記メモリから連続して並ぶデータとして上記現画面のブロックの画素及び上記参照画面のブロックの画素を読み出して残差を求めるようにしたことを特徴とする動きベクトル算出方法。

【請求項 2】 上記連続して並ぶ複数のデータに対して 1 命令で同様の処理を行なわせる命令を使って、上記残差の演算をするようにした請求項 1 に記載の動きベクトル算出方法。

【請求項 3】 上記現画面の画素及び上記参照画面の画素を市松模様状に N 画素毎に取り出すようした請求項 1 に記載の動きベクトル算出方法。

【請求項 4】 処理の対象となる現画面において分割された現画面のブロックに対して、参照画面内における同位置を基点としてそのブロックと大きさの等しい参照画面のブロックを抽出し、

所定の探索範囲内で上記参照画面のブロックを動かしながら、上記現画面のブロックと参照画面のブロックとの間の残差を求め、

上記残差が最小となる参照画面のブロックを検出して動きベクトルを算出する動きベクトル算出プログラムであって、

上記現画面の画素及び上記参照画面の画素を N 画素 (N は整数) 毎に取り出し、

上記 N 画素毎に間引かれた現画面の画素及び上記参照画面の画素を連続して並ぶデータとしてメモリに記憶させ、

上記メモリから連続して並ぶデータとして上記現画面のブロックの画素及び上記参照画面のブロックの画素を読み出して残差を求めるようにした動きベクトル算出プログラムを記録した記録媒体。

【請求項 5】 上記連続して並ぶ複数のデータに対して 1 命令で同様の処理を行なわせる命令を使って、上記残差の演算をするようにした請求項 4 に記載の動きベクトル算出プログラムを記録した記録媒体。

【請求項 6】 上記現画面の画素及び上記参照画面の画

素を市松模様状に N 画素毎に取り出すようした請求項 4 に記載の動きベクトル算出プログラムを記録した記録媒体。

【請求項 7】 処理の対象となる現画面において分割された現画面のブロックに対して、参照画面内における同位置を基点としてそのブロックと大きさの等しい参照画面のブロックを抽出し、

所定の探索範囲内で粗い精度で上記参照画面のブロックを動かしながら上記現画面のブロックと参照画面のブロックとの間の残差を求め、

上記残差が最小となる参照画面のブロックを検出して粗いサーチの動きベクトルを算出し、

上記粗いサーチで求められた動きベクトルの周辺で、より細かい精度で上記参照画面のブロックを動かしながら上記現画面のブロックと参照画面のブロックとの間の残差を求め、

上記残差が最小となる参照画面のブロックを検出してより細かい精度の動きベクトルを検出していくようにした動きベクトル算出方法において、

上記現画面及び上記参照画面の画素を第 1 のメモリに記憶させると共に、

上記現画面の画素及び上記参照画面の画素を N 画素 (N は整数) 毎に取り出し、

上記 N 画素毎に取り出された現画面の画素及び上記参照画面の画素を連続して並ぶデータとして第 2 のメモリに記憶させ、

上記粗いサーチの動きベクトルを算出する際には、上記第 2 のメモリに記憶されている N 画素毎に取り出され連続して並ぶデータとされた現画面の画素及び上記参照画面の画素を用い、

上記より細かい精度の動きベクトルを算出する際には、上記第 1 のメモリに記憶されている上記現画面及び上記参照画面の画素を用いるようにしたことを特徴とする動きベクトル算出方法。

【請求項 8】 上記連続して並ぶ複数のデータに対して 1 命令で同様の処理を行なわせる命令を使って、上記残差の演算をするようにした請求項 7 に記載の動きベクトル算出プログラムを記録した記録媒体。

【請求項 9】 上記現画面の画素及び上記参照画面の画素を市松模様状に N 画素毎に取り出すようした請求項 7 に記載の動きベクトル算出プログラムを記録した記録媒体。

【請求項 10】 処理の対象となる現画面において分割された現画面のブロックに対して、参照画面内における同位置を基点としてそのブロックと大きさの等しい参照画面のブロックを抽出し、

所定の探索範囲内で粗い精度で上記参照画面のブロックを動かしながら上記現画面のブロックと参照画面のブロックとの間の残差を求め、

上記残差が最小となる参照画面のブロックを検出して粗

いサーチの動きベクトルを算出し、
 上記粗いサーチで求められた動きベクトルの周辺で、より細かい精度で上記参照画面のブロックを動かしながら上記現画面のブロックと参照画面のブロックとの間の残差を求め、
 上記残差が最小となる参照画面のブロックを検出してより細かい精度の動きベクトルを検出していくようにした動きベクトル算出プログラムであって、
 上記現画面及び上記参照画面の画素を第1のメモリに記憶させると共に、
 上記現画面の画素及び上記参照画面の画素をN画素（Nは整数）毎に取り出し、
 上記N画素毎に取り出された現画面の画素及び上記参照画面の画素を連続して並ぶデータとして第2のメモリに記憶させ、
 上記粗いサーチの動きベクトルを算出する際には、上記第2のメモリに記憶されているN画素毎に取り出され連続して並ぶデータとされた現画面の画素及び上記参照画面の画素を用い、
 上記より細かい精度の動きベクトルを算出する際には、
 上記第1のメモリに記憶されている上記現画面及び上記参照画面の画素を用いるようにした動きベクトル算出プログラムを記録した記録媒体。

【請求項11】 上記連続して並ぶ複数のデータに対して1命令で同様の処理を行なわせる命令を使って、上記残差の演算をするようにした請求項10に記載の動きベクトル算出プログラムを記録した記録媒体。

【請求項12】 上記現画面の画素及び上記参照画面の画素を市松模様状にN画素毎に取り出すようにした請求項10に記載の動きベクトル算出プログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、例えば、MPEG (Moving Picture Experts Group) 2方式のエンコード処理をソフトウェアで行なう場合に用いて好適な動きベクトル算出方法及び動きベクトル算出プログラムを記録した記録媒体に関する。

【0002】

【従来の技術】 画像の高効率圧縮方式としてMPEG 2方式が普及している。MPEG 2方式は、動き補償予測符号化とDCT (Discrete Cosine Transform) とによりビデオ信号を圧縮符号化するものである。

【0003】 MPEG 2方式では、I (Intra) ピクチャと、P (Predicti) ピクチャと、B (Bidirectionally Predictive) ピクチャと呼ばれる3種類の画面が送られる。Iピクチャでは、同一のフレームの画素を使ってDCT符号化が行なわれる。Pピクチャでは、既に符号化されたIピクチャ又はPピクチャを参照して、動き補償予測を用いたDCT符号化が行なわれる。Bピクチャ

では、その前後のIピクチャ又はPピクチャを参照して、動き予測を用いたDCT符号化が行なわれる。

【0004】 なお、Pピクチャ及びBピクチャでは、マクロブロックの単位では、イントラ符号化を含む場合もある。すなわち、直流分を多く含むような画面では、画面内のイントラ符号化による方が効率的に圧縮できる。このような画面の場合には、イントラ符号化が行なわれる。

【0005】 図11は、従来のMPEG 2方式のエンコーダの構成の一例を示すものである。図11において、入力端子101に、例えば、輝度信号Yと、色差信号Cb、Crからなるコンポーネントデジタルビデオ信号が供給される。入力端子101からのデジタルビデオ信号は、動きベクトル検出回路103を介してフレームメモリ102に一旦蓄積される。このフレームメモリ102は、少なくとも、現画面と、過去の参照画面と、未来の参照画面との3フレーム分の画像を蓄積できる容量を有している。

【0006】 フレームメモリ102に蓄積されたデータを用いて、動きベクトル検出回路103で、参照画面と現画面との間の動きベクトルが求められる。動きベクトルMVは、例えば、(16×16)画素からなるマクロブロックを単位として求められる。求められた動きベクトルMVは、可変長符号化回路108、動き補償回路115に供給される。また、動きベクトル検出回路103で動きベクトルを求める際に得られる残差情報eは、モード設定回路104に供給される。

【0007】 フレームメモリ102の出力は、動きベクトル検出回路103を介して、スイッチ回路105の端子105Aに供給されると共に、減算回路106及び減算回路107に供給される。減算回路106の出力がスイッチ回路105の端子105Bに供給される。減算回路107の出力がスイッチ回路105の端子105Cに供給される。

【0008】 スイッチ回路105は、モード設定回路104からのモード設定信号により、符号化モードに応じて切り換えられる。すなわち、スイッチ回路105は、フレーム内符号化モードのときには、端子105A側に設定され、順方向予測符号化モードのときには、端子105B側に設定され、双方向予測符号化モードのときには、端子105C側に設定される。

【0009】 Iピクチャを伝送するときには、フレーム内予測符号化が行なわれる。このときには、スイッチ回路105は、端子105A側に設定される。そして、フレームメモリ102からは、現フレームの画像データが出力される。この現フレームの画像データは、スイッチ回路105を介して、DCT回路106に供給される。

【0010】 DCT回路106により、(8×8)を1ブロックとして、時間領域のビデオ信号がDCT変換され、周波数領域のスペクトラムデータに変換される。こ

のスペクトラムデータは、所謂ジグザグスキャンにより、直流分の成分から順に読み出される。DCT変換回路106の出力が量子化回路107に供給される。

【0011】量子化回路107には、送信バッファ109から送られてくる出力ビットレートの情報に応じて、量子化スケールが設定される。量子化回路107で、この量子化スケールにより、DCT変換回路106からのスペクトラムデータが量子化される。これにより、出力されるビットストリームのビットレートが一定に保たれる。

【0012】量子化回路107の出力は、可変長符号化回路108に供給されると共に、逆量子化回路110に供給される。可変長符号化回路108で、量子化回路107からのスペクトラムデータは、動きベクトルと、量子化スケールと、予測モードと共に、可変長符号化される。

【0013】可変長符号化回路108からは、符号化されたビットストリームが出力される。このビットストリームは、送信バッファ109に一旦蓄積される。送信バッファ109からは、要求されるビットレートに応じてデータストリームが読み出され、出力端子111から、ビットストリームが出力される。

【0014】また、量子化回路107の出力が逆量子化回路110に供給される。逆量子化回路110の出力がIDCT回路112に供給される。Iピクチャの場合には、フレーム内の画素でDCT変換が行なわれるため、逆量子化回路110及びIDCT回路112により、元の画面が形成される。この1画面分のデジタルビデオデータは、加算回路113を介して、画像フレームメモリ114に供給される。このフレームメモリ114に蓄積されたデータが次のPピクチャ又はBピクチャでの参照フレームのデータとされる。

【0015】Pピクチャを伝送するときには、参照フレームとの間で順方向予測符号化が行なわれる。このときには、スイッチ回路105は、端子105B側に設定される。フレームメモリ102からは、現フレームのデジタルビデオ信号が出力される。このフレームメモリ102の出力は、減算回路106に供給される。

【0016】フレームメモリ114には、参照フレームのデータが蓄積されている。この参照フレームのデータは、動き補償回路115により、動き補償され、減算回路106に供給される。減算回路106で、現フレームのデータと、動き補償された参照フレームのデータとの差分が求められる。この現フレームのデータと、参照フレームのデータとの差分データがスイッチ回路105を介して、DCT回路106に供給される。

【0017】DCT回路106により、参照フレームのデータと現フレームのデータとの差分データがDCT変換され、スペクトラムデータに変換される。DCT変換回路106の出力が量子化回路107に供給される。量

量子化回路107で、DCT変換回路106からのスペクトラムデータが量子化される。

【0018】量子化回路107の出力は、可変長符号化回路108に供給されると共に、逆量子化回路110に供給される。可変長符号化回路108で、量子化回路107からのスペクトラムデータは、動きベクトルと、量子化スケールと、予測モードと共に、可変長符号化される。

【0019】可変長符号化回路108からは、符号化されたビットストリームが出力される。このビットストリームは、送信バッファ109に一旦蓄積される。送信バッファ109からは、要求されるビットレートに応じてデータストリームが読み出され、出力端子111から、ビットストリームが出力される。

【0020】また、量子化回路107の出力が逆量子化回路110に供給される。逆量子化回路110の出力がIDCT回路112に供給される。Pピクチャの場合には、参照フレームのデータと現画面のデータとの差分がDCT変換されるため、逆量子化回路110及びIDCT回路112により、参照フレームのデータと現画面のデータとの差分が得られる。この差分データが加算回路113に供給される。

【0021】加算回路113には、フレームメモリ114から、動き補償回路115を介して、それまでの参照画面のデータが供給される。加算回路113で、それまでの参照画面のデータに差分データが加算される。この加算回路113の出力が次の参照フレームのデータとしてフレームメモリ114に蓄積される。

【0022】Bピクチャを伝送するときには、過去の参照フレームと、未来の参照フレームとを使って、両方向予測符号化が行なわれる。このときには、スイッチ回路105は、端子105C側に設定される。フレームメモリ102からは、現フレームのデジタルビデオ信号が出力される。このバッファメモリ102の出力は、減算回路107に供給される。

【0023】フレームメモリ114には、過去参照フレームと、未来の参照フレームのデータが蓄積されている。この両方の参照フレームのデータは、動き補償回路115により、動き補償され、減算回路107に供給される。減算回路107で、現フレームのデータと、動き補償された過去及び未来の参照フレームのデータとの差分が求められる。この現フレームのデータと、過去及び未来の参照フレームのデータとの差分データがスイッチ回路105を介して、DCT回路106に供給される。

【0024】DCT回路106により、過去及び未来の参照フレームのデータと、現フレームのデータとの差分データがDCT変換され、スペクトラムデータに変換される。DCT変換回路106の出力が量子化回路107に供給される。量子化回路107で、DCT変換回路106からのスペクトラムデータが量子化される。

【0025】量子化回路107の出力は、可変長符号化回路108に供給されると共に、逆量子化回路110に供給される。可変長符号化回路108で、量子化回路107からのスペクトラムデータは、動きベクトルと、量子化スケールと、予測モードと共に、可変長符号化される。

【0026】可変長符号化回路108からは、符号化されたビットストリームが出力される。このビットストリームは、送信バッファ109に一旦蓄積される。送信バッファ109からは、要求されるビットレートに応じてデータストリームが読み出され、出力端子111から、ビットストリームが出力される。

【0027】

【発明が解決しようとする課題】近年、CPU (Central Processing Unit) の処理速度が飛躍的に向上されると、大容量のメモリが安価に入手できるようになってきている。そこで、上述のようなMPEG2のエンコード処理をソフトウェアで行なうことが考えられている。

【0028】ところが、MPEG2のエンコードには、動きベクトルを算出する処理が必要である。動きベクトルは、処理の対象となる現フレームにおいて分割されたブロックに対して、参照フレーム画面内における同位置を基点としてそのブロックと大きさの等しいブロックを抽出し、所定の探索範囲内でこのブロックを動かしながら、参照フレームのブロックの各画素と現フレームのブロックの各画素との差分の絶対値和から残差を求め、この残差が最小となる参照フレームのブロックを求めるようなブロックマッチングにより求められる。このブロックマッチングの処理は、多くの演算が必要であり、このことがMPEG2のエンコード処理をソフトウェアで行なわせる場合に問題となる。

【0029】つまり、図12において、現フレーム201のブロックCBLKでの動きベクトルを求める場合には、このブロックCBLKの位置を基点とする参照フレーム202の周囲が探索範囲SAとされる。この探索範囲SA内の参照フレームのブロックRBLKが取り出され、このブロックRBLKの(16×16)の各画素と、現フレームのブロックCBLKの(16×16)の各画素との間の差分が求められ、この差分の絶対値和から、残差が求められる。参照フレーム203のブロックRBLKは、所定の探索範囲SA内で動かされ、所定の探索範囲SA内のブロックRBLK内の各位置において、同様に、ブロックRBLKの各画素と、現フレーム201のブロックCBLKの各画素との間の差分が求められ、この差分の絶対値和から、残差が求められる。これら各位置で求められた参照フレームのブロックRBLKの各画素と、現フレームのブロックCBLKの各画素との間の差分の絶対値和が比較され、残差が最小となるブロックがマッチングブロックとされる。このマッチングブロックから動きベクトルが求められる。

【0030】このようなブロックマッチングから動きベクトルを検出する場合、ブロックとして例えば(16×16)を用いると、各画素の差分を求めるのに、 $16 \times 16 = 256$

回の減算が必要であり、これらの各画素の差分の絶対値和を求めるのに、256の数の総和を求める加算が必要である。

【0031】更に、1画素ステップで所定の探索範囲内で参照ブロックを動かしながら動きベクトルを検出すると、その探索範囲の画素に相当する回数だけ残差を求める処理が必要になる。したがって、1画素ステップで所定の探索範囲内でブロックを動かしながら残差を求め、この残差が最小になるブロックの位置から動きベクトルを検出するようにすると、その演算量は膨大となり、ソフトウェアで処理するには難しくなる。

【0032】このような動きベクトルの探索を高速化するためには、二つのアプローチが考えられる。一つは、1回のブロックマッチングの計算量を削減することであり、他の一つは、探索領域内でのブロックマッチングの回数を減らすことである。前者の方法の一つとして、参照フレーム及び現フレームのブロック内の画素を市松模様状に間引いてブロックマッチングを行なうものがある。このように、ブロックを市松模様状に間引くと、絶対値差分和の演算回数を半減させることができる。

【0033】ところが、ブロックを市松模様状に間引くと、画素のデータが連続しなくなるため、MMX命令が使えなくなる。つまり、MMX命令は、連続して並ぶ複数のデータに対して1命令で同様の処理を行なわせる命令であり、マルチメディアを扱える近年のパーソナルコンピュータでは、MMX機能を有するCPUを備えたものが多い。ブロックマッチングでは、画素間の差分の絶対値和を求める処理が行なわれるので、MMX命令が有効に使えと、演算速度が高速化できる。ところが、ブロックを市松模様状に間引いてしまうと、画素のデータが連続しなくなる。このため、連続して並ぶ複数のデータに対して1命令で同様の処理を行なわせるようなMMX命令を使って処理ができなくなる。このため、ブロックを市松模様状に間引いてブロックマッチングの回数を減らしたとしても、処理時間は殆ど短縮化しないことになる。

【0034】したがって、この発明の目的は、探索領域内でのブロックマッチングの回数を減らして高速処理を図ると共に、MMX命令を有効に活用できるようにした動きベクトル算出方法及び動きベクトル算出プログラムを記録した記録媒体を提供することにある。

【0035】

【課題を解決するための手段】この発明は、処理の対象となる現画面において分割された現画面のブロックに対して、参照画面内における同位置を基点としてそのブロックと大きさの等しい参照画面のブロックを抽出し、所

定の探索範囲内で参照画面のブロックを動かしながら、現画面のブロックと参照画面のブロックとの間の残差を求め、残差が最小となる参照画面のブロックを検出して動きベクトルを算出する動きベクトル算出方法において、現画面の画素及び参照画面の画素をN画素（Nは整数）毎に取り出し、N画素毎に取り出された現画面の画素及び参照画面の画素を連続して並ぶデータとしてメモリに記憶させ、メモリから連続して並ぶデータとして現画面のブロックの画素及び参照画面のブロックの画素を読み出して残差を求めるようにしたことを特徴とする動きベクトル算出方法及びこのような動きベクトル算出プログラムが記録された記録媒体である。

【0036】この発明は、処理の対象となる現画面において分割された現画面のブロックに対して、参照画面内における同位置を基点としてそのブロックと大きさの等しい参照画面のブロックを抽出し、所定の探索範囲内で粗い精度で参照画面のブロックを動かしながら現画面のブロックと参照画面のブロックとの間の残差を求め、残差が最小となる参照画面のブロックを検出して粗いサーチの動きベクトルを算出し、粗いサーチで求められた動きベクトルの周辺で、より細かい精度で参照画面のブロックを動かしながら現画面のブロックと参照画面のブロックとの間の残差を求め、残差が最小となる参照画面のブロックを検出してより細かい精度の動きベクトルを検出していくようにした動きベクトル算出方法において、現画面及び参照画面の画素を第1のメモリに記憶させると共に、現画面の画素及び参照画面の画素をN画素（Nは整数）毎に取り出し、N画素毎に取り出された現画面の画素及び参照画面の画素を連続して並ぶデータとして第2のメモリに記憶させ、粗いサーチの動きベクトルを算出する際には、第2のメモリに記憶されているN画素毎に取り出され連続して並ぶデータとされた現画面の画素及び参照画面の画素を用い、より細かい精度の動きベクトルを算出する際には、第1のメモリに記憶されている現画面及び参照画面の画素を用いるようにしたことを特徴とする動きベクトル算出方法及びこのような動きベクトル算出プログラムが記録された記録媒体である。

【0037】参照フレーム及び現フレームのブロック内の画素を市松模様状に間引いてブロックマッチングを行なうようにしている。この時、市松模様状に取り出された現フレームの画素及び参照フレームの画素は、連続して並ぶデータとしてメモリに記憶される。このようにすると、ブロックマッチングの際にMMX命令を有効に利用することができ、高速化が図れる。

【0038】また、現フレームの画素及び参照フレームの画素が記憶されるメモリと、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとして記憶されるメモリとが用意される。そして、まず、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとして

記憶されたメモリを使って、2画素ステップのサーチが行なわれる。この時、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶させているので、メモリ上で1つずつ参照ブロックを動かしていくと、2画素ステップでサーチしていくことになる。2画素ステップのサーチにより動きベクトルが求められたら、求められた点の周辺で、現フレームの画素及び参照フレームの画素の全画素が記憶されたメモリを使って、1画素ステップのサーチが行なわれる。

【0039】このように、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶させるようにすると、2画素ステップでサーチしていくことになるため、MMX命令が使えるだけでなく、ログリズミックサーチが簡単に行なえるという利点が生じる。

【0040】

【発明の実施の形態】以下、この発明の実施の形態について図面を参照して説明する。図1は、この発明が適用できるデータ処理装置の構成を示すものである。

【0041】図1において、1はCPU（Central Processing Unit）、2はROM（Read Only Memory）、3はRAM（Random Access Memory）である。CPU1、ROM2、RAM3は、プロセッサバス4に接続される。

【0042】CPU1としては、例えば、MMX機能を有するプロセッサが用いられる。MMX機能は、動画の再生や、画像の加工、音声合成のような処理を高速で行なえるようにしたものである。すなわち、MMX命令では、SIMD（Single Instruction Multiple Data）と呼ばれる技術を用いることにより、1回の命令で連続したデータに対して同じ処理を繰り返して行なえる。

【0043】ROM2には、ブートストラップのプログラムが格納されている。RAM3は、作業用のメインメモリで、RAM3の容量としては、例えば、64Mbit以上のものを搭載することが望まれる。

【0044】CPU1は、ブリッジ回路5に接続されており、ブリッジ回路5からプロセッサバス4が導出される。ブリッジ回路5は、PCI（Peripheral Component Interconnect）バス6に接続される。ブリッジ回路5は、CPU1と、プロセッサバス4及びPCIバス6とを繋ぐものである。

【0045】PCIバス6には、IDE（Integrated Device Electronics）コントローラ7、SCSI（Small Computer System Interface）コントローラ8、グラフィックアクセラレータ9、IEEE（Institute Of Electrical and Electronics Engineers）1394コントローラ10が接続される。

【0046】IDEコントローラ7には、ハードディスクドライブやCDドライブ等のストレージデバイス11

が接続される。SCSIコントローラ8には、ハードディスクドライブやCDドライブ等のストレージデバイス12が接続される。また、SCSIコントローラ8には、ストレージデバイスの他、イメージスキャナ等、種々の周辺機器を取り付けることができる。グラフィックスアクセラータ9には、ディスプレイ13が取り付けられる。IEEE1394コントローラ10には、デジタルVCR (Video Cassette Recorder) 等のデジタルオーディオビデオ機器を接続することができる。

【0047】PCIバス6は、ブリッジ回路14を介して、ISA (Industrial Standard Architecture) バス15に接続される。ブリッジ回路14は、PCIバス6とISAバス15とを繋ぐものである。ISAバス15には、入力デバイスコントローラ16、フロッピディスクコントローラ17、パラレルコントローラ18、RS232Cコントローラ19が接続される。

【0048】入力デバイスコントローラ16には、キーボードやマウス等の入力デバイス20が接続される。フロッピディスクコントローラ17には、フロッピディスクドライブ21が接続される。パラレルコントローラ18には、プリンタ等を取り付けることができる。RS232Cコントローラ19には、モデム等を取り付けることができる。

【0049】初期状態では、まず、ROM2に格納されているブートストラップのプログラムが走り、初期設定が行なわれる。そして、ストレージデバイス11又は12がアクセスされ、ストレージデバイス11又は12にインストールされているオペレーティングシステムが読み込まれ、プログラムの常駐部がメインメモリとされたRAM3に常駐される。これにより、オペレーティングシステムが起動され、このオペレーティングシステムの管理の基に、種々の処理が実行される。

【0050】なお、上述の例では、PCIバスとISAバスとを用いた構成とされているが、このような構成に限定されるものではない。USB (Universal Serial Bus) を設け、このUSBバスにキーボードやマウス等の種々の機器を接続するようにしても良い。

【0051】上述のようなデータ処理装置でMPEG2のエンコード処理を行なう場合には、MPEG2のエンコード処理を行なうアプリケーションプログラムが実行される。このアプリケーションプログラムは、IDEのハードディスク等のストレージデバイス11又はSCSIのハードディスク等のストレージデバイス12に実行可能なプログラムとして格納されており、このアプリケーションプログラムは、実行時にはRAM3に読み込まれ、CPU1により逐次実行される。

【0052】なお、このMPEG2のエンコード処理を行なうアプリケーションプログラムは、IDEのハードディスク等のストレージデバイス11又はSCSIのハードディスク等のストレージデバイス12に予めインス

トールしておくようにしても良いし、CD-ROMやフロッピディスクにより提供するようにしても良い。更に、通信で提供するようにしても良い。

【0053】このMPEG2のエンコード処理を行なうアプリケーションプログラムが実行されると、デジタルビデオデータに対して、予測モードに応じて、動きベクトルの算出処理、DCT演算処理、量子化処理、可変長符号化処理が行なわれ、デジタルビデオデータがMPEG2方式により圧縮される。このとき、作業用として、RAM3が用いられ、演算は、CPU1の演算機能を用いて行なわれる。デジタルビデオデータは、例えば、IEEE1394コントローラ10に接続された外部のデジタルVCR等から入力される。そして、出力データは、例えば、SCSIコントローラ8やIDEコントローラ7に接続されたハードディスクドライブ等に記録される。

【0054】図2は、このようなMPEG2のエンコード処理プログラムの概要を示すフローチャートである。

【0055】図2に示すように、まず、複数フレームのデジタルビデオデータが取り込まれ、このデジタルビデオデータがRAM3上にバッファされる (ステップS1)。そして、ブロックマッチングにより、動きベクトルが算出される (ステップS2)。

【0056】予測モードがIピクチャか、Pピクチャか、Bピクチャかが判断される (ステップS3)。ステップS3でIピクチャであると判断されたら、同一フレーム内での (8×8) 画素を1ブロックとしてDCT演算が行なわれる (ステップS4)。そして、求められた係数データは量子化され (ステップS5)、可変長符号化される (ステップS6)。また、このときのデータは、参照画像のデータとしてRAM3上に保存される (ステップS7)。

【0057】ステップS3でPピクチャであると判断されたら、順方向の参照画像のデータがRAM3から読み出され (ステップS8)、この参照画像に対して、ステップS2で算出された動きベクトルに基づいて動き補償がなされ (ステップS9)、現画像のデータと、動き補償された参照画像のデータの差分が求められ、この現画像のデータと、参照画像のデータの差分データがDCT演算される (ステップS10)。そして、求められたデータは量子化され (ステップS11)、可変長符号化される (ステップS12)。また、このときのデータは、参照画像のデータとしてRAM3上に保存される (ステップS13)。

【0058】ステップS3でBピクチャであると判断されたら、両方向の参照画像のデータがRAM3から読み出され (ステップS14)、この参照画像に対して、ステップS2で算出された動きベクトルに基づいて動き補償がなされ (ステップS15)、現画像のデータと、動き補償された参照画像のデータの差分が求められ、こ

の現画像のデータと、参照画像のデータとの差分データがDCT演算される(ステップS16)。そして、求められたデータは量子化され(ステップS17)、可変長符号化される(ステップS18)。

【0059】上述のステップS2で示す動きベクトルの算出処理は、処理の対象となる現フレームにおいて分割されたブロックに対して、参照フレームの画面内における同位置を基点としてそのブロックと大きさの等しいブロックを抽出し、所定の探索範囲内でこのブロックを動かしながら、現フレームのブロックの画素と参照フレームのブロックの各画素との差分の絶対値和から残差を求め、この残差が最小となるブロックを検出するようなブロックマッチングにより行なわれる。ところが、このようなブロックマッチングにより動きベクトルを求めると、その演算量は膨大となる。

【0060】そこで、この発明では、図3に示すよう、参照フレーム及び現フレームのブロック内の画素を市松模様状に間引いてブロックマッチングを行なうようしている。

【0061】図3において、参照フレームのブロック31は(16×16)画素からなり、この(16×16)画素の中から、市松模様状に(8×16)画素が取り出される。また、現フレームのブロック31は(16×16)画素からなり、この(16×16)画素の中から、市松模様状に(8×16)画素が取り出される。

【0062】そして、この時、MMX命令を有効に利用してブロックマッチングが行なえるように、この市松模様状に取り出された現フレームの画素及び参照フレームの画素は、連続して並ぶデータとしてメモリ(RAM3の所定の領域)に記憶させる。

【0063】すなわち、図4Aに示すように、現フレームの画素及び参照フレームの画素から市松模様状に画素が取り出される。このように間引かれた画素は、図4Bに示すように、データが連続するように再配置され、図4Cに示すように、メモリの連続するアドレス上に取り込まれる。

【0064】このように連続して並ぶデータとして現フレームの画素及び参照フレームの画素をメモリに記憶させると、MMX命令を使ってブロックマッチングの処理が行なえるようになるため、高速な演算処理が可能となる。

【0065】また、このように、市松模様状に間引かれた現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶させるようにすると、2画素ステップでサーチしていくことになるため、MMX命令が使えただけでなく、ログリズミックサーチが簡単に行なえるという利点が生じる。

【0066】ここで、ログリズミックサーチは、最初に探索範囲内において粗くサーチを行って残差が最小となる点を検出し、次に、その点の周囲で精度を上げてサー

チを行なって残差が最小となる点を検出して動きベクトルを求めるようなものである。

【0067】市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶されている場合には、以下のようにしてログリズミックサーチが行なわれる。

【0068】現フレームの画素及び参照フレームの画素の全画素が記憶されたメモリと、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとして記憶されたメモリ(又はメモリ領域)とが用意される。そして、先ず、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとして記憶されたメモリを使って、2画素ステップの粗いサーチが行なわれる。2画素ステップの粗いサーチにより動きベクトルが求められたら、求められた点の周辺で、現フレームの画素及び参照フレームの画素の全画素が記憶されたメモリを使って、1画素ステップのより細かいサーチが行なわれる。これにより、最終的な動きベクトルが求められる。

【0069】例えば、図5Aに示すように、各フレームの画像データF1、F2、F3、…が入力されるとする。そして、この入力画像F1、F2、F3、…をエンコードして、図5Bに示すように、I、B、B、P、B、B、P、…ピクチャの順でMPEG2のピクチャP1、P2、P3、…をエンコードするとする。

【0070】このようなエンコード処理において、上述のようにして動きベクトルを求めるために、図6に示すように、作業用のRAM3には、入力された1フレーム分の画像データの全画素を夫々蓄積するメモリ領域21A~21Fと、1フレーム分の画素を市松模様状に取り出して連続したデータとして蓄積するメモリ領域22A~22Cが設けられる。

【0071】図7に示すように、画像データF1、F2、F3、…が入力されると、この画像データは、1フレーム毎にメモリ領域21A~21Fに保存されると共に、この1フレーム分の画像データの中から、市松模様状に1サンプル毎の画素が取り出され、この市松模様状に取り出された画素は、連続するアドレスに再配置され、画像データf1、f2、f3、…としてメモリ領域22A~22Cに保存される。

【0072】つまり、時点T1で画像データF1がメモリ領域21Aに取り込まれ、時点T2で画像データF2がメモリ領域21Bに取り込まれ、時点T3で画像データF3がメモリ領域21Cに取り込まれ、時点T4で画像データF4がメモリ領域21Dに取り込まれる。

【0073】そして、時点T4では、画像データF1から市松模様状に1サンプル毎画素が取り出され、連続したアドレスに配列されたデータf1がメモリ領域22Aに蓄積される。また、画像データF4から市松模様状に1サンプル毎画素が取り出され、連続したアドレスに配

列されたデータ f 4 がメモリ領域 22B に蓄積される。

【0074】時点 T5 で、画像データ F5 がメモリ領域 21E に蓄積される。また、画像データ F2 から市松模様状に 1 サンプル毎画素が取り出され、連続したアドレスに配列されたデータ f 2 がメモリ領域 22C に蓄積される。

【0075】時点 T6 で、画像データ F6 がメモリ領域 21F に蓄積される。また、画像データ F3 から市松模様状に 1 サンプル毎の画素が取り出され、連続したアドレスに配列されたデータ f 3 がメモリ領域 22C に蓄積される。

【0076】時点 T7 で、画像データ F7 がメモリ領域 21A に蓄積される。また、画像データ F7 から市松模様状に 1 サンプル毎画素が取り出され、連続したアドレスに配列されたデータ f 7 がメモリ領域 22A に蓄積される。

【0077】以下、図 7 に示すように、各時点で入力されたフレームの画像データがメモリ領域 21A~21F に蓄積されると共に、各フレームの画像データから市松模様状に 1 サンプル毎画素が取り出され、連続したアドレスに配列されたデータがメモリ領域 22A~22C に蓄積される。

【0078】このように、メモリ領域 21A~21F に保存された画像データ F1、F2、F3、…及びメモリ領域 22A~22C に保存された市松模様状に間引かれた画像データ f1、f2、f3、…を用いて、動きベクトルが求められる。動きベクトルは、2 画素ステップで所定の探索範囲のサーチを行い、この 2 画素ステップのサーチで求められた点の周辺を、1 画素ステップでサーチするようなログリズミックサーチにより求められる。

【0079】ピクチャ P1 は、I ピクチャであるので、時点 T1~T3 の間でエンコードすることができる。

【0080】時点 T4 で、P ピクチャであるピクチャ P4 のエンコード処理が行なわれ、その動きベクトルが求められる。ピクチャ P4 では、参照フレームとして F1 が用いられ、現フレームとして F4 が用いられる。この場合、最初の 2 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 22A に蓄積されているデータ f1 が用いられ、現フレームのブロックとして、メモリ領域 22B に蓄積されているデータ f4 が用いられる。次の 1 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 21A に蓄積されているデータ F1 が用いられ、現フレームのブロックとして、メモリ領域 21D に蓄積されているデータ F4 が用いられる。

【0081】時点 T5 で、B ピクチャであるピクチャ P2 のエンコード処理が行なわれ、その動きベクトルが求められる。ピクチャ P2 では、参照フレームとして F1 と F4 が用いられ、現フレームとして F2 が用いられる。この場合、最初の 2 画素ステップのサーチは、参照

フレームのブロックとして、メモリ領域 22A に蓄積されているデータ f1 と、メモリ領域 22B に蓄積されているデータ f4 とが用いられ、現フレームのブロックとして、メモリ領域 22C に蓄積されているデータ f2 が用いられる。次の 1 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 21A に蓄積されているデータ F1 と、メモリ領域 21D に蓄積されているデータ F4 とが用いられ、現フレームのブロックとして、メモリ領域 21B に蓄積されているデータ F2 が用いられる。

【0082】時点 T6 で、B ピクチャであるピクチャ P3 のエンコード処理が行なわれ、その動きベクトルが求められる。ピクチャ P3 では、参照フレームとして F1 と F4 が用いられ、現フレームとして F3 が用いられる。この場合、最初の 2 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 22A に蓄積されているデータ f1 と、メモリ領域 22B に蓄積されているデータ f4 とが用いられ、現フレームのブロックとして、メモリ領域 22C に蓄積されているデータ f3 が用いられる。次の 1 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 21A に蓄積されているデータ F1 と、メモリ領域 21D に蓄積されているデータ F4 とが用いられ、現フレームのブロックとして、メモリ領域 21C に蓄積されているデータ F3 が用いられる。

【0083】時点 T7 で、P ピクチャであるピクチャ P7 のエンコード処理が行なわれ、その動きベクトルが求められる。ピクチャ P7 では、参照フレームとして F4 が用いられ、現フレームとして F7 が用いられる。この場合、最初の 2 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 22B に蓄積されているデータ f4 が用いられ、現フレームのブロックとして、メモリ領域 22A に蓄積されているデータ f7 が用いられる。次の 1 画素ステップのサーチは、参照フレームのブロックとして、メモリ領域 21D に蓄積されているデータ F4 が用いられ、現フレームのブロックとして、メモリ領域 21A に蓄積されているデータ F7 が用いられる。

【0084】以下、同様に、時点 T8 で、B ピクチャであるピクチャ P5 の動きベクトルが求められ、時点 T9 で、B ピクチャであるピクチャ P6 の動きベクトルが求められる。

【0085】図 8 は、上述のように、ログリズミックサーチにより動きベクトルを算出する再の処理を示すフローチャートである。図 8 において、入力画像データが保存されると共に、参照フレーム及び現フレームのデータから市松模様状に 1 サンプルおきの画素データが抽出され、この 1 サンプルおきの画素データが連続したアドレス上に再配置されて保存される（ステップ S21）。

【0086】1 画面の全ブロックについての処理が終了

したか否かが判断される（ステップS22）。

【0087】1画面の全ブロックの処理が終了していなければ、最初に、所定の探索範囲内で2画素ステップでブロックを動かしながらブロックマッチングのサーチが行なわれる。（ステップS23）。

【0088】2画素ステップの動きベクトルが検出されたら、次に、その動きベクトルの周辺で、1画素ステップでブロックを動かしながらブロックマッチングのサーチが行なわれる（ステップS24）。

【0089】このようにして求められた結果が保存され（ステップS25）、次のブロックに進められる（ステップS26）。そして、ステップS22で、このブロックが最後のブロックか否かが判断され、最後のブロックでなければ、同様の処理が繰り返され、次のブロックでの動きベクトルが求められる。画面全体のブロックの動きベクトルが求められたら、ステップS22で、最後のブロックであると判断され、処理が終了される。

【0090】図9は、図8におけるステップS23の2画素ステップのサーチの処理を示すフローチャートである。この2画素ステップサーチ処理は、現フレームの画素及び参照フレームの画素が市松模様状に取り出され、連続して並ぶデータとして記憶されたメモリを使って行なわれる。

【0091】図9において、まず、探索範囲の開始点が決められる（ステップS31）。そして、縦方向の探索開始位置が最上端にリセットされる（ステップS32）。縦方向において、最下端の探索位置まで到達したか否かが判断され（ステップS33）、最下端でなければ、横方向の位置が最左端にリセットされる（ステップS34）。

【0092】探索範囲の右端に達したか否かが判断される（ステップS35）。探索範囲の右端に達していなければ、市松模様状の（8×16）のブロックでのブロックマッチング処理が行なわれ、残差が求められる（ステップS36）。

【0093】このときの残差ADがそれまでの最小値ADminより小さいか否かが判断され（ステップS37）、残差ADが最小値ADminより小ければ、今回の残差ADが最小値ADminとされ、動きベクトルMVが現在の位置とされて（ステップS38）、横方向の位置が2画素ステップ動かされる（ステップS39）。なお、現フレームの画素及び参照フレームの画素が市松模様状に取り出され、連続して並ぶデータとしてメモリに記憶させているので、2画素ステップ動かすことは、メモリ上では1ステップずつ動かすことに相当する。

【0094】ステップS37で、残差ADがそれまでの最小値ADminより小さくなければ、ステップS39に行き、横方向の位置が2画素ステップ動かされる。そして、ステップS35にリターンされる。

【0095】ステップS35で、探索範囲の右端に達し

たか否かが判断され、探索範囲の右端に達していなければ、同様の処理が繰り返される。これにより、探索範囲を左から右にブロックを動かしながら、残差が求められ、それまでの最小の残差が最小値ADminとして保存される。

【0096】ステップS35で、探索範囲の右端に達したと判断されると、縦方向に2画素ステップだけブロックが動かされる（ステップS40）。そして、ステップS33にリターンされる。そして、以下、同様の処理が実行される。

【0097】縦方向において、最下端の探索位置まで到達すると、ステップS33で、最下端の探索位置まで到達したと判断される。最下端の位置まで到達したと判断されたら、その結果が動きベクトルMVとして保存される（ステップS41）。なお、この動きベクトルMVは、次の1画素ステップでのサーチでの基準点となる。

【0098】図10は、図8におけるステップS24の1画素ステップのサーチの処理を示すフローチャートである。この1画素ステップのサーチ処理は、現フレーム及び参照フレームの全画素が記憶されたメモリを使って行なわれる。

【0099】図10において、まず、図9におけるステップS41で求められた次のステップのサーチの基準点の左上に開始点が決められる（ステップS51）。そして、縦方向の探索開始位置が最上端にリセットされる（ステップS52）。縦方向において、最下端の探索位置まで到達したか否かが判断され（ステップS53）、最下端でなければ、横方向の位置が最左端にリセットされる（ステップS54）。

【0100】探索範囲の右端に達したか否かが判断される（ステップS55）。探索範囲の右端に達していなければ、（16×16）のブロックでのブロックマッチング処理が行なわれ、残差が求められる（ステップS56）。

【0101】このときの残差ADがそれまでの最小値ADminより小さいか否かが判断され（ステップS57）、残差ADが最小値ADminより小ければ、今回の残差ADが最小値ADminとされ（ステップS58）、動きベクトルMVが現在の位置とされて、横方向の位置が1画素ステップ動かされる（ステップS59）。

【0102】ステップS57で、残差ADがそれまでの最小値ADminより小さくなければ、ステップS59に行き、横方向の位置が1画素ステップ動かされる。そして、ステップS55にリターンされる。

【0103】ステップS55で、探索範囲の右端に達したか否かが判断され、探索範囲の右端に達していなければ、同様の処理が繰り返される。これにより、探索範囲を左から右にブロックを動かしながら、残差が求められ、それまでの最小の残差が最小値ADminとして保

存される。

【0104】ステップS55で、探索範囲の右端に達したと判断されると、縦方向に1画素ステップだけブロックが動かされる(ステップS60)。そして、ステップS53にリターンされる。そして、以下、同様の処理が実行される。

【0105】縦方向において、最下端の探索位置まで到達すると、ステップS53で、最下端の探索位置まで到達したと判断される。最下端の位置まで到達したと判断されたら、その結果から動きベクトルMVが求められ、

10 処理が終了する。

【0106】なお、上述の例では、参照フレーム及び現フレームの1サンプル毎に市松模様状に取り出しているが、間引き方はこれに限定されるものではない。間引き間隔や、間引き方については、各種のものが適用可能である。

【0107】また、上述の例では、ログリズミックサーチを行なう際に、最初の粗いサーチで、1サンプル毎に間引かれた画素が蓄積されたメモリ上で、アドレスを1ずつ動かすことにより、2画素ステップのサーチを行なっているが、アドレスを2ずつ動かすことにより、4画素ステップのサーチを行なったり、アドレスを3ずつ動かすことにより、6画素ステップのサーチを行なったりするようにしても良い。また、上述の例では、2画素ステップのサーチと、1画素ステップのサーチとからログリズミックサーチを行なっているが、更に、複数段のログリズミックサーチを行なうようにしても良い。

【0108】

【発明の効果】この発明によれば、参照フレーム及び現フレームのブロック内の画素を市松模様状に間引いてブ

ロックマッチングを行なうようにしている。この時、市松模様状に取り出された現フレームの画素及び参照フレームの画素は、連続して並ぶデータとしてメモリに記憶される。このようにすると、ブロックマッチングの際にMMX命令を有効に利用することができ、高速化が図れる。

【0109】また、現フレームの画素及び参照フレームの画素が記憶されるメモリと、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとして記憶されるメモリとが用意される。そして、

40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60

まず、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとして記憶されたメモリを使って、2画素ステップのサーチが行なわれる。この時、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータ

タとしてメモリに記憶させているので、メモリ上で1つずつ参照ブロックを動かしていくと、2画素ステップでサーチしていくことになる。2画素ステップのサーチにより動きベクトルが求められたら、求められた点の周辺で、現フレームの画素及び参照フレームの画素の全画素が記憶されたメモリを使って、1画素ステップのサーチが行なわれる。

【0110】このように、市松模様状に取り出された現フレームの画素及び参照フレームの画素を連続して並ぶデータとしてメモリに記憶させるようにすると、2画素ステップでサーチしていくことになるため、MMX命令が使えるだけでなく、ログリズミックサーチが簡単に行なえるという利点が生じる。

【図面の簡単な説明】

【図1】データ処理装置の一例のブロック図である。

【図2】MPEG2のエンコード処理の説明に用いるフローチャートである。

【図3】市松模様状の間引き処理の説明に用いる略線図である。

20 【図4】市松模様状のデータを連続したデータとして並べる際の説明に用いる略線図である。

【図5】この発明が適用されたMPEG2エンコーダでのエンコード処理の説明に用いる略線図である。

【図6】この発明が適用されたMPEG2エンコーダでのエンコード時のメモリ構成の説明に用いる略線図である。

【図7】この発明が適用されたMPEG2エンコーダでのエンコード時の説明に用いるタイミング図である。

30 【図8】この発明が適用されたMPEG2エンコーダでの動きベクトルの算出処理の説明に用いるフローチャートである。

【図9】この発明が適用されたMPEG2エンコーダでの動きベクトルの算出処理の説明に用いるフローチャートである。

【図10】この発明が適用されたMPEG2エンコーダでの動きベクトルの算出処理の説明に用いるフローチャートである。

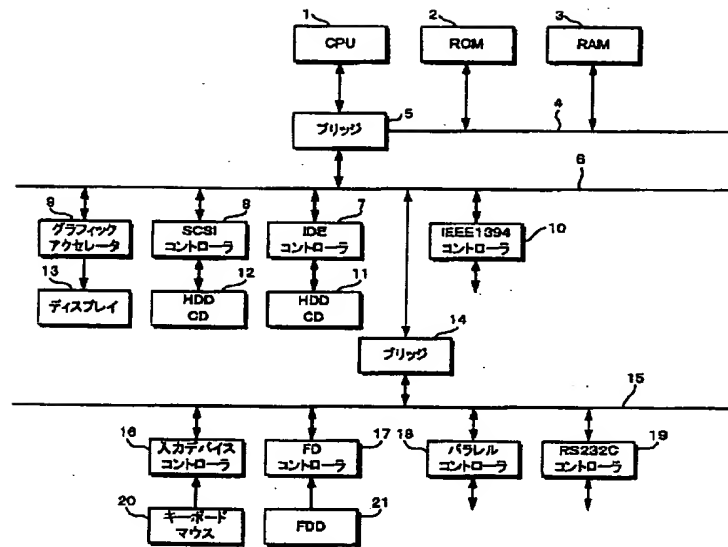
【図11】従来のMPEG2エンコーダの構成を示すブロック図である。

40 【図12】ブロックマッチングの説明に用いる略線図である。

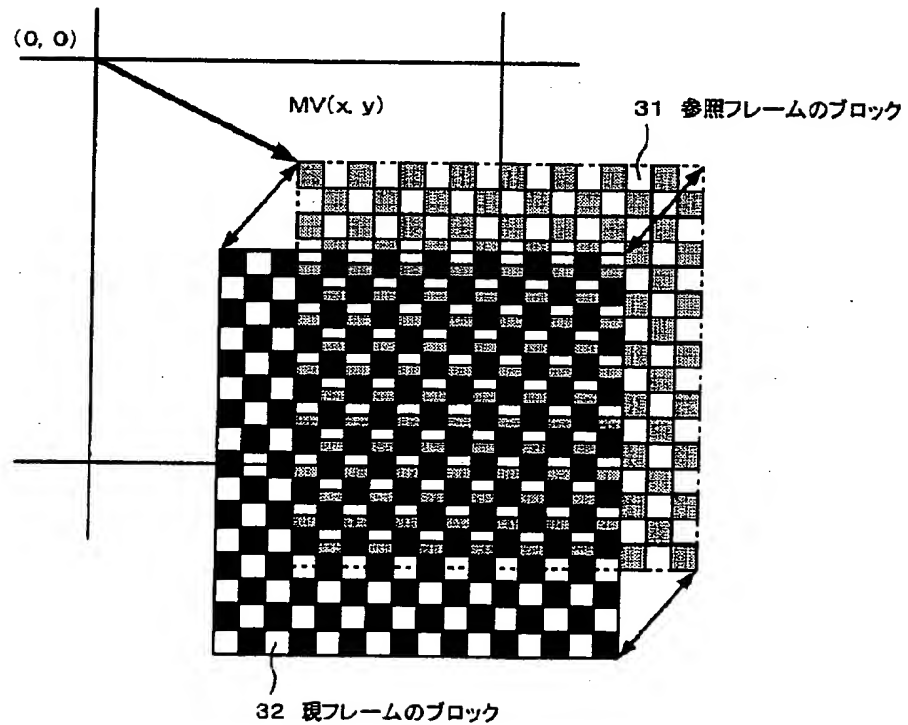
【符号の説明】

1・・・CPU, 2・・・ROM, 3・・・RAM, 7
・・・IDEコントローラ, 8・・・SCSIコントローラ, 10・・・IEEE1394コントローラ

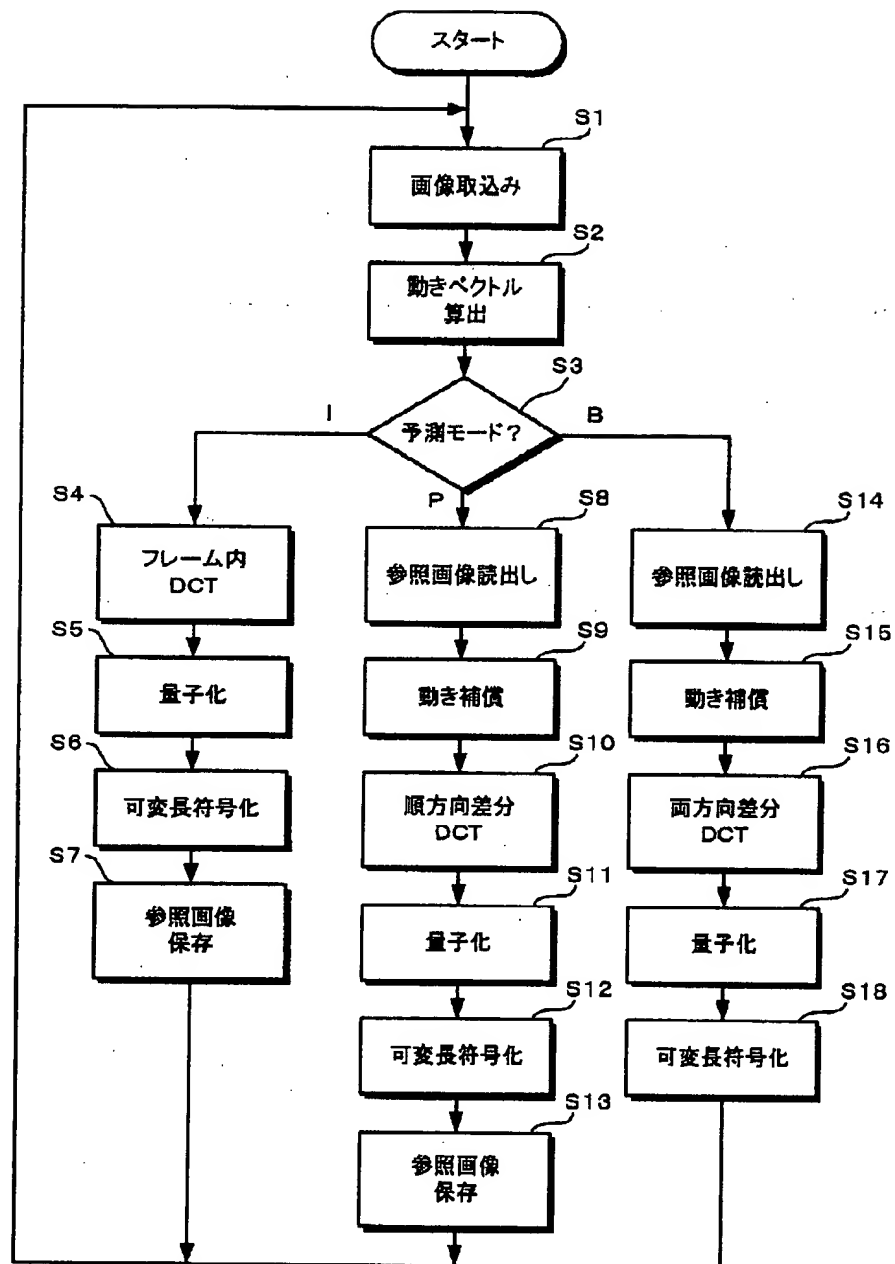
【図 1】



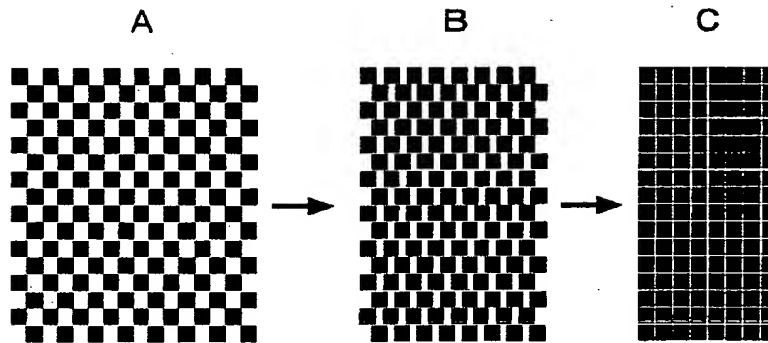
【図 3】



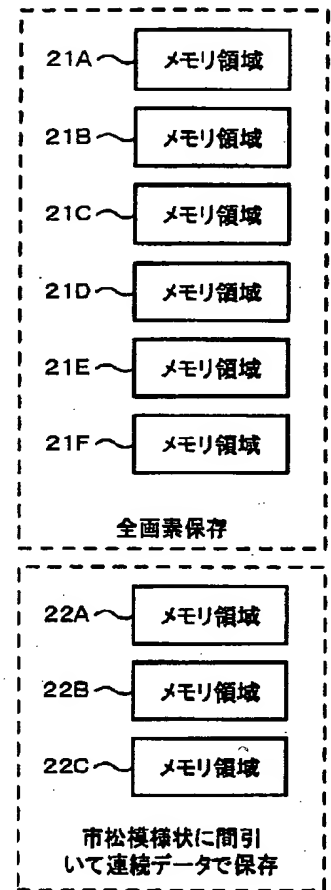
【図2】



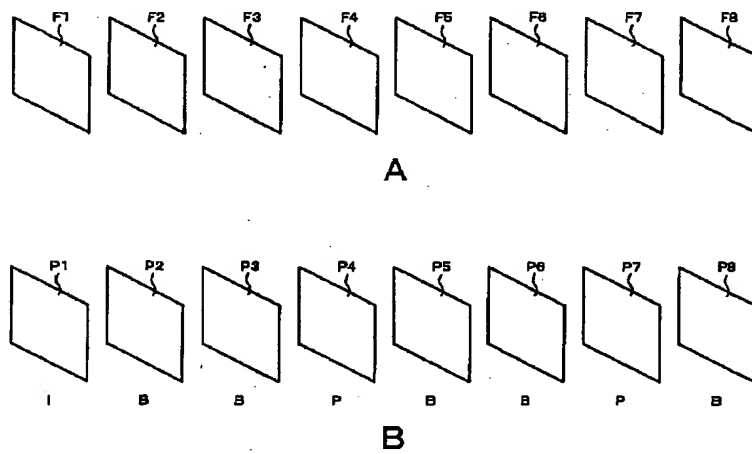
【図4】



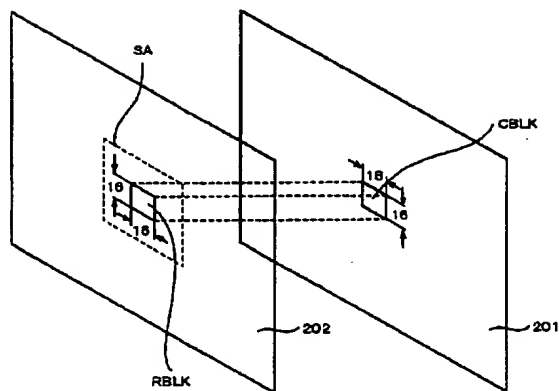
【図6】



【図5】



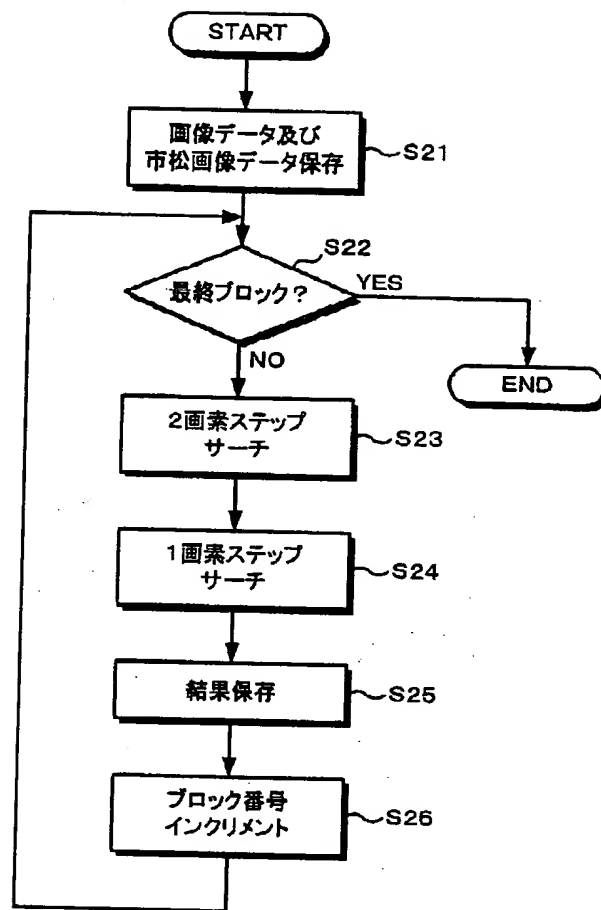
【図12】



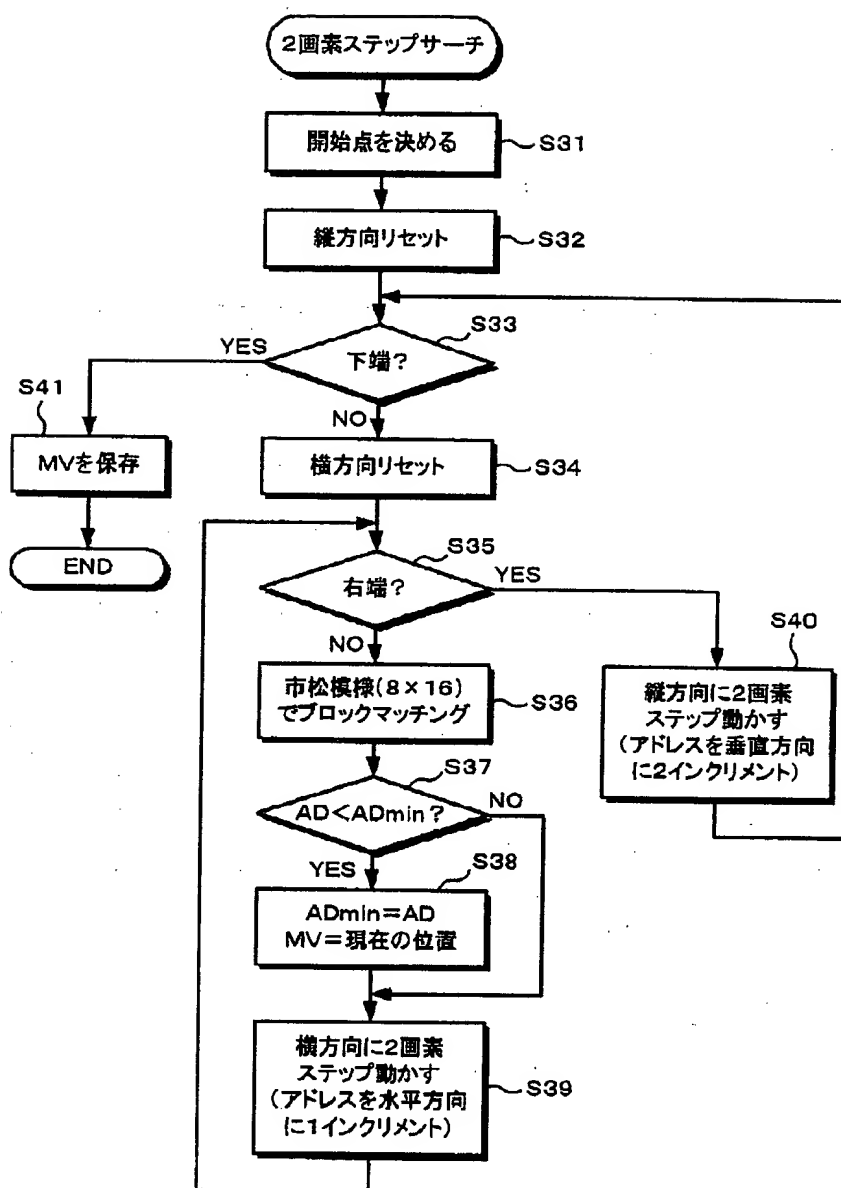
【図7】

時点	T1	T2	T3	T4	T5	T6	T7	T8	T9
メモリ領域 21A	F1	F1	F1	F1	F1	F1	F7	F7	F7
メモリ領域 21B		F2	F2	F2	F2	F2	F2	F8	F8
メモリ領域 21C			F3	F3	F3	F3	F3	F3	F9
メモリ領域 21D				F4	F4	F4	F4	F4	F4
メモリ領域 21E					F5	F5	F5	F5	F5
メモリ領域 21F						F6	F6	F6	F6
メモリ領域 22A				f1	f1	f1	f7	f7	f7
メモリ領域 22B				f4	f4	f4	f4	f4	f4
メモリ領域 22C					f2	f3	f3	f5	f6
動作	P1 エンコード			P4 エンコード	P2 エンコード	P3 エンコード	P7 エンコード	P5 エンコード	P6 エンコード

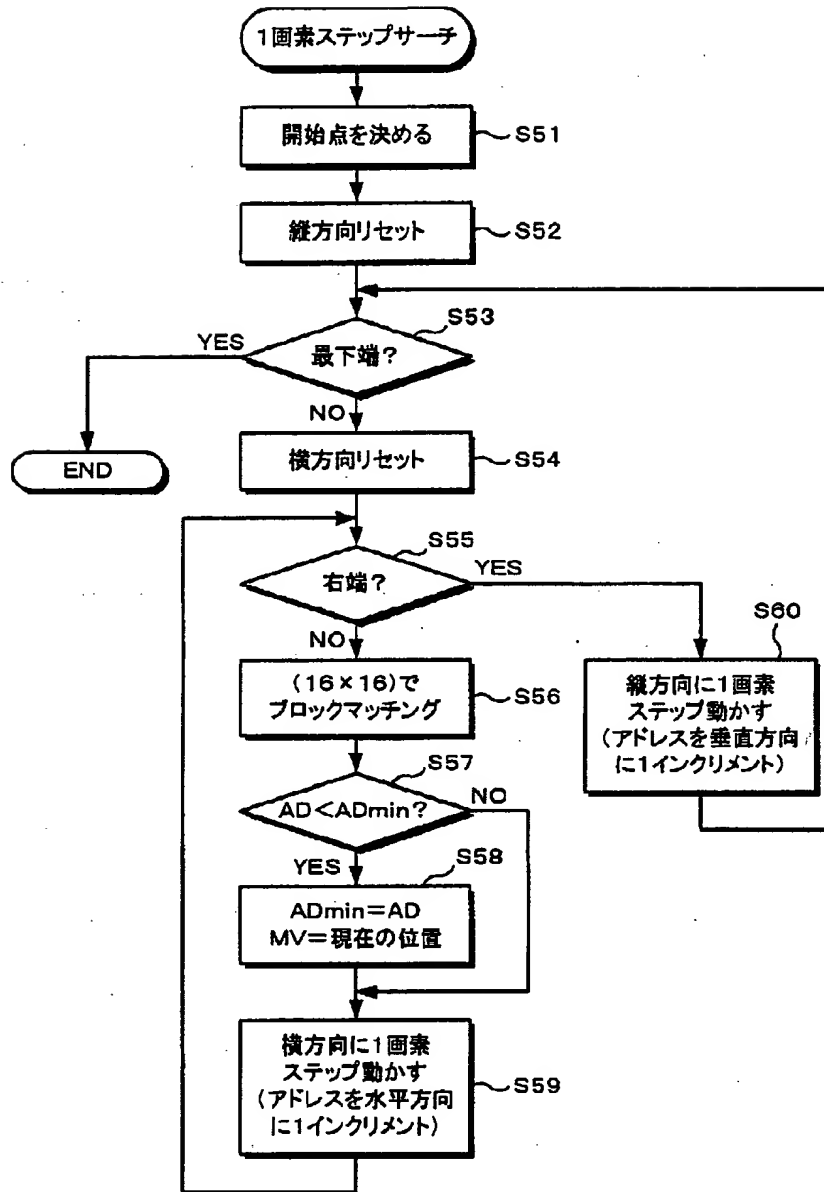
【図8】



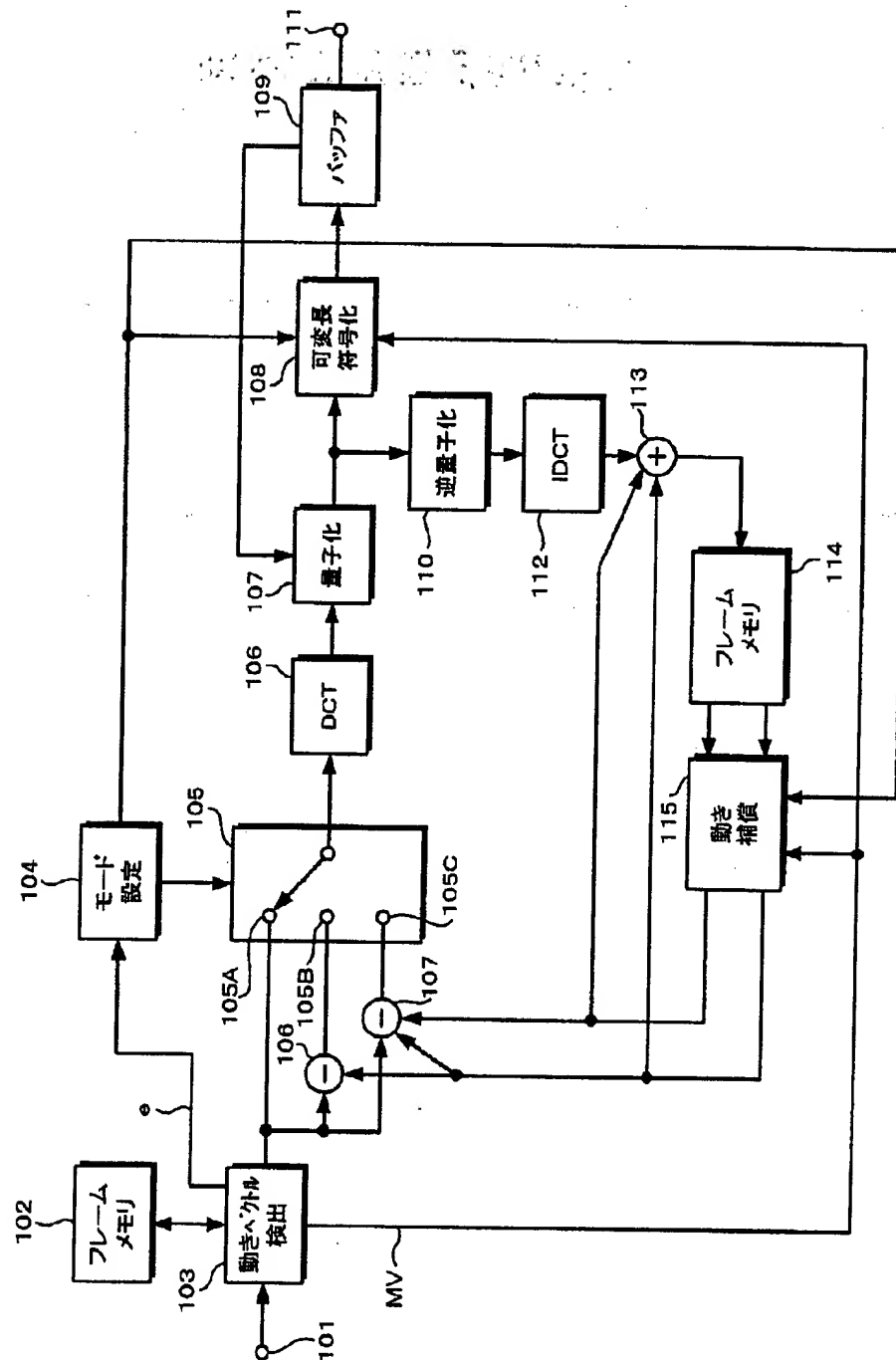
【図9】



【図10】



【図 11】



THIS PAGE BLANK (USPTO)